DOI: 10.1609/aaai.12006

SPECIAL TOPIC ARTICLE

Clarity 2.0: Improved assessment of product competitiveness from online content

Yufeng Huang[1] | **Mariana Bernagozzi**[2] | **Michelle Morales**[2] | **Sheema Usmani**[2] | **Biplav Srivastava**[3] | **Michelle Mullins**[2]

[1] Sacred Heart University, Fairfield, CT, USA

[2] IBM, Armonk, NY, USA

[3] University of South Carolina, Columbia, SC, USA

Correspondence
Yufeng Huang, West Campus East Building WCE*E-1124, Sacred Heart University, 5151 Park Ave., Fairfield, CT 06825, USA.
Email: yufeng.chase.huang@gmail.com

Abstract

Competitive analysis is a critical part of any business. Product managers, sellers, and marketers spend time and resources scouring through an immense amount of online and offline content, aiming to discover what their competitors are doing in the marketplace to understand what type of threat they pose to their business' financial well-being. Currently, this process is time and labor-intensive, slow and costly. This paper presents *Clarity*, a data-driven unsupervised system for assessment of products, which is currently in deployment in the global technology company, IBM. *Clarity* has been running for more than a year and is used by over 4,500 people to perform over 200 competitive analyses involving over 1000 products. The system considers multiple factors from a collection of online content: numeric ratings by online users, sentiment of user generated online content for key product performance dimensions, content volume, and topic analysis of content. The results and explanations of factors leading to the results are visualized in an interactive dashboard that allows users to track their product's performance as well as understand main contributing factors. Its efficacy has been tested in a series of cases across IBM's portfolio which spans software, hardware, and services. After initial release and first year of use, improvements to the methodology were implemented to ensure it was relevant to and served the highest impact needs of target users. Moreover, new use cases leveraging the initial ideas and approaches continue to be explored.

OVERVIEW

Every business wants to know how their product/offering performs relative to competition. Competitive analysis is a critical process for many roles, particularly for marketers, sellers, and product managers. Currently, such users scan through the large volume of online and offline content, aiming to understand what competitors are doing in the marketplace for every product they have, to understand what type of threats they may pose to the business's position in the market. This process is time and labor-intensive, error-prone, slow and costly. Furthermore, as competition and feedback from users continue to evolve, any previous analysis needs to be frequently updated to stay relevant.

To address this business need, we introduce a deployed system, called *Clarity*, which analyzes the competitive landscape of products in a marketplace continuously at scale, as data is updated over time. This article is an

© 2021, Association for the Advancement of Artificial Intelligence. All rights reserved.

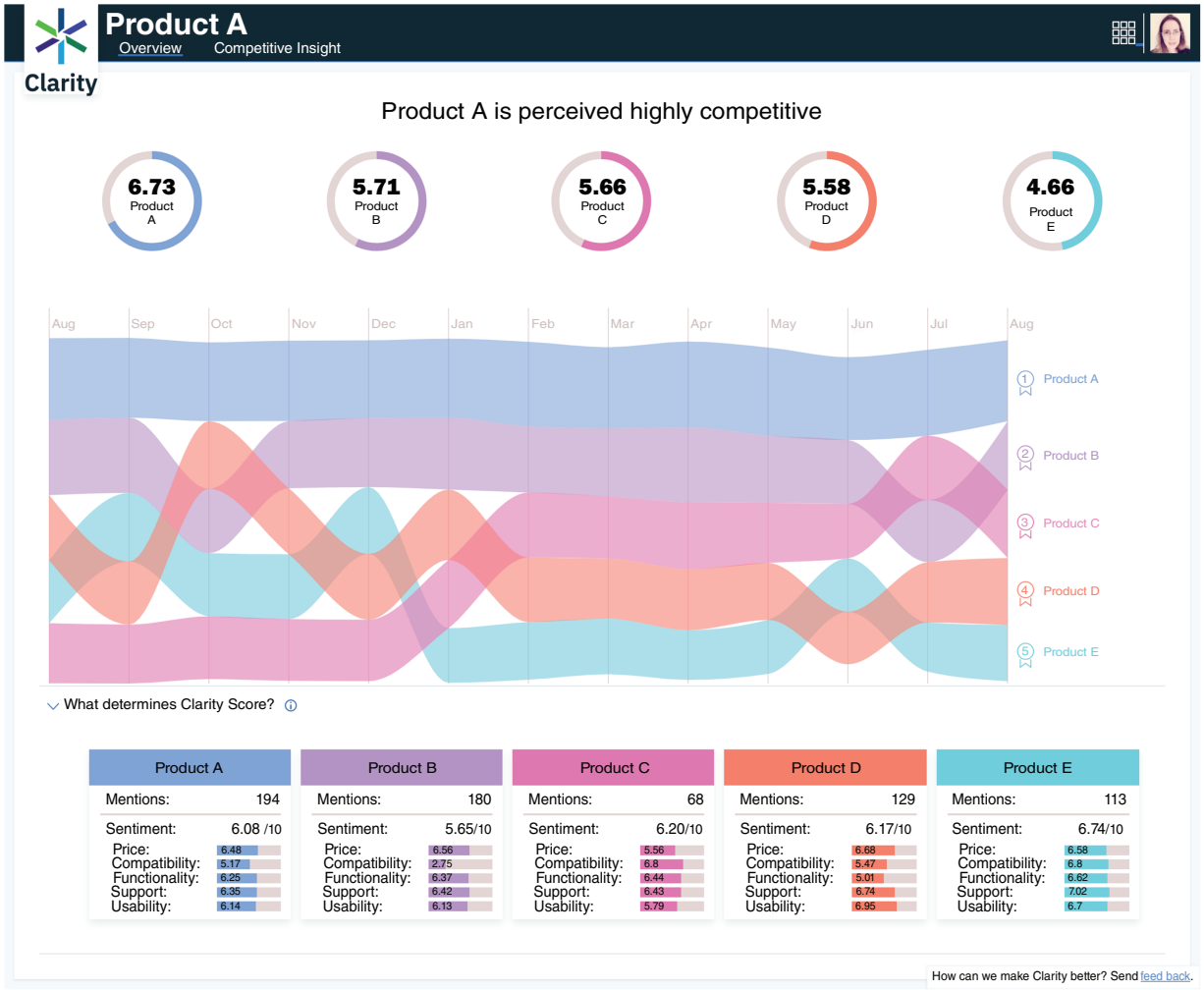

FIGURE 1 Sorted stream graph to visualize products competitiveness

extension of our original IAAI-20 (Innovative Applications of Artificial Intelligence) paper, which presented the first iteration of our deployed system (Usmani et al. 2020). We will provide updates on the latest developments in the system, highlighting how we have improved model performance with a new NLP model to extract product information and new scoring method to compare products. With new experiments, we show that the improvements lead to an overall better user experience and significant realization of business value. By *Clarity*, we will refer to the latest system but where necessary, we will refer to the two versions of the systems as *Clarity* 1.0 and *Clarity* 2.0, respectively. We will now first preview the working of the system by providing a use case example.

Use case example

Let us consider the competitive landscape for Product-A. We first determine the similar products which Product A competes with. In this example, they are referred to as Products B, C, D and E. The selection of products for a marketplace is a business decision. The output of *Clarity* is visualized in Figure 1.

All the products are compared based on the *Clarity Score*, which is a numerical value summarizing the online contents. The *Clarity Score* for each product is displayed through a visualization, as shown at the top part of Figure 1. The charts provide a comparison of different products based on the numerical values.

In Figure 1, a ranking of the products are plotted to provide a quantitative overview of the competitiveness of the target products with respect to competitors over 12 months. Using different color schemes for different products, the ranking over the predefined time period is displayed. In addition to the ranking, the width for each product represents the normalized Clarity Score in that period. As we can see in the chart, Product A has the highest score over the time period considered, thus it was ranked first throughout the chart. However, the ranking could change

dramatically across time. For example, Product D (shown in orange) was ranked third at the beginning of the time period, then it was ranked fourth in the following month, and then the ranking changed again to second. With this information, the stakeholders of the target product can get a sense of how all the players are performing in the market.

To shed light on which factors are contributing to the Product's score, more details about how the *Clarity Score* is calculated for each product are shown in the bottom of Figure 1. For each product, the main contributors to the score are the *number of mentions* and the overall *Sentiment score*. The sentiment score is the aggregated value of the 5 drivers of the product. As we can see, this gives a more granular level of information of how the products are compared. For example, although Product A has an overall higher score than Product B, Product B receives a higher average *Sentiment score* in its price. However, Product B has a much lower *Compatibility* driver score, which is the main contributor to its lower overall score.

Current users of *Clarity* use the score in their workflow to understand the competitive stance of their product in the marketplace and leverage the detailed factor analysis to understand their products' strengths/weaknesses as well as those of their competition. Together the high level and detailed level analysis help users make data-driven, informed, decisions regarding the strategic development plan of their products.

The remainder of the paper is organized as follows: we start with the background and related work, then provide a succinct system overview of the original system. Next, we discuss the improvements made and present experiments to evaluate their benefits. Finally, we conclude with discussion on new use cases and future work.

In the middle section of the graph, X-axis represents time, Y-axis corresponds to product rank and the thickness of the line corresponds to absolute competitiveness score.

BACKGROUND

In this section, we will discuss the competitive analysis process and related effort so that the contribution of our work and the impact of our system can be better understood.

Related work

There is a large-scale trend of leveraging artificial intelligence to improve efficiency and outcomes for business operations like business development (Srivastava et al. 2018), marketing, sales, and product development. Furthermore, Natural Language Processing (NLP) methods, including text mining, are being used to understand many parts of the business landscape including customer needs, product competitiveness, and company performance. Specifically, researchers have surveyed the area of competitive intelligence for products and have demonstrated the promise of approaches using NLP and text mining (Amarouche, Benbrahim, and Kassou 2015).

In Joung et al. (2018), the authors use text mining methods to analyze customer complaints and find gaps in the company's products. In Afful-Dadzie et al. (2014), the authors perform text analysis on user comments posted on social media to compare telecommunication providers in Ghana. In (Bhatt, Mcneil, and Patel 2014), the authors track general sentiment overtime for products by calculating a sentiment score based on user-generated content such as reviews and comments.

The work presented in Usmani et al. (2020) builds upon previous work by introducing a novel competitive metric that encompasses sentiment as one of its contributing factors. Our system not only provides a metric but also aims to explain performance, which is a critical step in the market intelligence process. To the best of our knowledge, Clarity is the first unsupervised approach for ranking and assessment of product competitiveness.

CLARITY 1.0: SYSTEM OVERVIEW

The architecture, capabilities, processes, operations, and visualizations of *Clarity 1.0* is discussed extensively in our previous paper (Usmani et al. 2020). The improvements made to the system since the publication of the aforementioned article are described in detail in sections *Improved NLP Model in Clarity 2.0* and *Improved Clarity Score in Clarity 2.0*.

We start with some basic concepts and notations: (1) a set of products: p_1 to p_N., (2) a set of data sources: d_1 to d_M . These data sources are public forums and review sites, and (3) a set of documents, also called posts or reviews: d_1 to do. Each document d_k is associated to one product p_i and one data source d_j .

The main steps of *Clarity* are (1) to prepare online content of products p_1 to p_N from sources d_1 to d_M, and extract keywords and sentiments using NLP techniques (offline); (2) to process request for analysis for product pi and generate *Clarity Score* (online) and (3) to visualize analysis results using rich Data-driven documents (D3) (Bostock, Ogievetsky, and Heer 2011) (on-line, optional).

IMPROVED NLP MODEL IN CLARITY 2.0

In *Clarity*, we analyze public user commentary from various online sources on IBM and competitor products to understand the overall sentiment and topics of concern for users. In *Clarity* 1.0, online commentary is processed using machine learning techniques, and then fed into a series of downstream statistical analyses to provide a summary of important information and trends for our users. Because the downstream analyses are dependent on the first step of NLP on the input text, as mentioned in our previous paper (Usmani et al. 2020), it is necessary to keep improving the NLP algorithms to ensure that the later statistical analyses are performed on trust-worthy inputs.

In the first version of *Clarity*, we built a complex algorithm around IBM Watson Natural Language Understanding (NLU) to extract insights from keywords. Once the keywords were extracted, they were analyzed out of the context of the original content, and clustered into a predefined list of topics. Although we can perform appropriate statistical analyses on such a large corpus of keywords, the feedback from our users has indicated that keyword analyses out of sentence context could lead to inaccurate topic assignments, and thus undermining the analyses we present to our users. A common issue faced with AI applications is user skepticism of generated insights - which we experienced with our initial keyword based version. To iterate and improve, while simultaneously increasing user confidence in and adoption of *Clarity*, we shifted to a sentence based approach so that we could provide more trustworthy analyses.

The main task of NLP in *Clarity* is topic classification. By using sentences rather than keywords, we have a large set of techniques at our disposal. A simple approach is to perform topic modeling on the commentary to cluster the sentences based on co-occurrences of n-gram keywords. However, *Clarity* aims to align the specific analyses with our users. Rather than using a set of topics determined by the unsupervised topic modeling algorithms, we performed rigorous business research to come up with a list of 14 topics that are appropriate to our users. Because of the fixed list of topics, we have to collect labeled data to perform supervised machine learning to classify the sentences into the respective topics.

Topic classification is one of the most typical tasks in text analytics. Supervised learning methods for text classification is a two-step process. First, features are extracted from the input text, then a machine learning model is trained on these features and the corresponding labels.

Traditional feature engineering includes bag-of-words, bag of n-grams, and term frequency-inverse document frequency (TF-IDF). However, these count based methods are not able to capture the semantics and context of the input sentences.

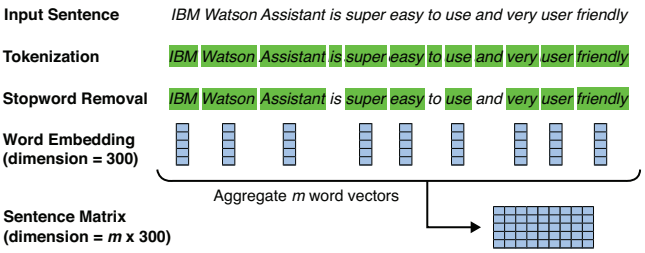

FIGURE 2 Example of text pre-processing for all the NLP algorithms discussed in this following sections

More advanced topic classification techniques start with vector embeddings of each individual words in the input text. BERT-embedding is one of the most popular context based embedding methods in recent years due to the use of self-attention. However, in our current case of topic classification on online content, BERT embedding poses a significant bottleneck due to its reliance on computing power, especially when a large amount of text is processed. In fact, unlike semantic analysis and other text understanding tasks where the order of words is important, topic classification on online content depends mostly on the presence of certain keywords. Feature extract on the word embedding level is sufficient for proper topic classification. To include some of the contextual information, topic level attention can be added.

In the following sections, we start from simple feature engineering on sentences, and then gradually include more advanced techniques to capture more information for better classification. For consistent comparison, the same text pre-processing and pre-trained word embedding, and neural networks with the same number of parameters are used for all of the following supervised machine learning approaches.

Text preprocessing

As discussed above, the models considered here are not trained on the sequence of words in the sentences. Thus, stop words and punctuations are removed prior to the application of word embedding. The process is illustrated in Figure 2.

First, the input sentence is tokenized into words. Then, the stop words are removed. Finally, the word embeddings are applied to the remaining words in the sentence and aggregated to a sentence matrix of dimension $m \times 300$, where m is the number of words after stop word removal.

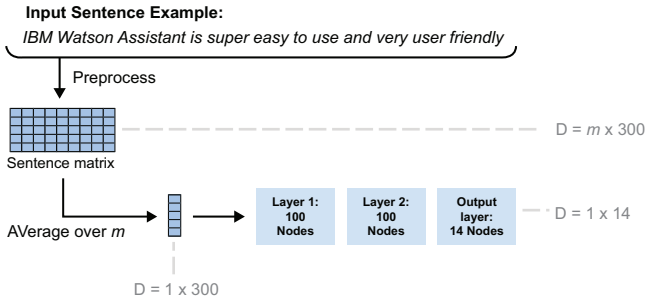

FIGURE 3 Architecture of the neural network on sentence

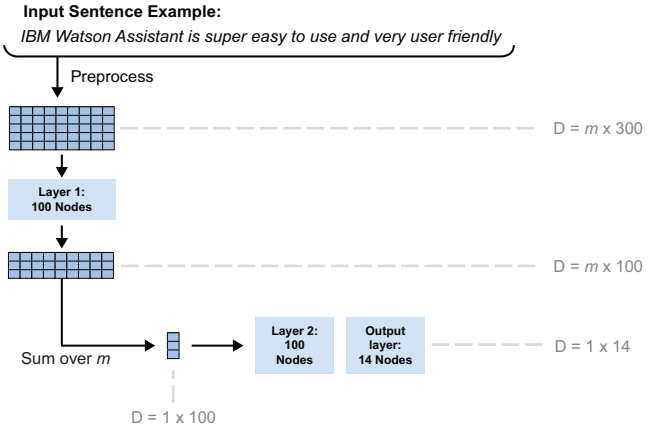

FIGURE 4 Architecture of the neural network on words

Neural networks on sentences

The simplest approach using word embeddings for machine learning is to separate the embedding part and the machine learning part. First, word vectors are obtained from a pre-trained word embedding model. Then, the word vectors for each sentence are averaged to obtain the sentence vector. Since pre-trained word embeddings are used, this part of the algorithm does not require further training.

Next, when sentence vectors are obtained, they are aligned with the corresponding labels obtained in the data collection part to feed into a fully connected multi-layered neural networks. The dimension of the input layer is 300, and the number of nodes in the first two layers are 100, and then 14 as the output layer, because the input text is classified into 14 pre-defined topics. The number of parameters in the model is $(300 \times 100 + 100) + (100 \times 100 + 100) + (100 \times 14 + 14) = 41614$.

The neural network layers are shown in Figure 3 as the shaded blue blocks. The dimension of the intermediate vectors, D, is shown to the right of the figure. m is the number of words in the sentence after stop word removal.

Neural networks on words

Because the neural network model was trained on the averaged sentence vector, some of the important features from keywords could be averaged out or hidden by the presence of other words. This issue could be alleviated by retraining the word embedding on the current corpus of all the scraped online content. However, to solve the problem directly, we adopt a different approach in which the first layer of the neural network is applied to the word vectors before aggregating. By doing this, the neural network can be trained to amplify the important dimensions and minimize the dimensions that contribute to noises. However, because important features are already extracted using this approach, rather than taking the average where the corresponding dimensions could be changed due to the number of words, the sum of the output of the first layer of the neural network is used.

The architecture of the corresponding neural network is as follows. The first layer of the neural network with 100 nodes is applied directly to the word vector, then the outputs are summed over words to produce a vector with a dimension of 100. Then the second layer of 100 nodes is applied, and lastly the output layer of the neural network has 14 nodes. This neural network architecture has the same number of parameters as the previous method. However, by moving the nodes closer to the words before summing, an NLP model is trained on the features of each individual words. Such a model is highly efficient because the first layer is shared between words.

The neural network layers are shown in Figure 4 as the shaded blue and pink blocks. The dimension of the intermediate vectors, D, is shown to the right of the vector. m is the number of words in the sentence after stop word removal.

Neural networks based on attention

Finally, the neural network can be further improved by applying the attention mechanism between topics and words, so that the words that important to a certain topic will have a larger contribution than the words that are not relevant.

The neural network layers are shown in Figure 5 as the shaded blue blocks. The dimension of the intermediate vectors, D, is shown to the right of the vector. m is the number of words in the sentence after stop word removal.

Similar to the neural networks on words, the first layer is applied to the output of word embedding. However, rather than summing over all the word vectors in a sentence, an attention layer is applied. Let K be the output from the first

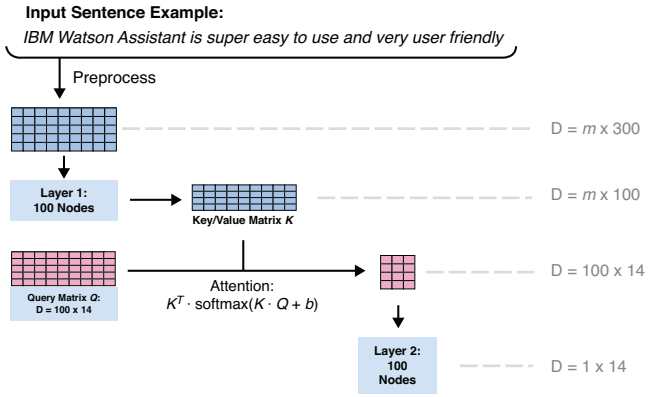

FIGURE 5 Architecture of the neural network based on attention

layer of neural network, then K has a dimension of $m \times 100$, where m is the number of words in the sentence. Let Q of dimension 100×14 and b of dimension 14 be trainable parameters, then the mechanism can be written mathematically as: $L = KT \cdot \text{softmax}(K \cdot Q + b)$, which has a final dimension of 100×14. Here, K is used as both the key matrix and value matrix, and Q is used as a trainable query matrix.

When the softmax function is applied to $KT \cdot Q + b$, words that are more relevant to a topic will be assigned a larger weight. After the application of the attention layer, the output can be fed to another fully connected neural network with 100 nodes, such that when summed in this dimension the output will become 14, corresponding to the 14 topics of interest. Since the first layer has 100 nodes, the attention layer contains trainable matrices of dimensions 100×14 and 14, and because the last layer contains 100 nodes, the total number of parameters is again the same as the former two methods described in the previous sections. However, the attention-based neural network not only outperforms the previous neural networks based on words, but also assigns a weight of each word to all the topics.

With the attention mechanism, the contribution of keywords to the topic classification can be calculated for interpretation. As shown in Figure 6, the sentence, *much easier to use than the competition and you will see nearly immediate results*, is a sample sentence from online posts. The sentence is classified into three topics, ease of use, performance and efficiency, and product competitiveness. By highlighting the words based on the corresponding to the attention weights, the relevant keywords are clearly identified for each topic. It is important to note that these keywords are completely learned by the algorithm during training, implying that the algorithm is appropriate for this task to match keywords similar to human intuition.

The example sentence is classified into three topics, ease of use, performance and efficiency, and product competitiveness, with corresponding probabilities, respectively. The darkness of the highlighted keyword is calculated by multiplying the attention weight with the classification probability, which reflects the contribution of the keyword to the classification of a certain topic.

Improved Clarity Score in Clarity 2.0

The main purpose of the *Clarity* project is to compare IBM products with competitor products. In order to compare these products over a wide array of features, a *Clarity Score* is calculated to rank products in the same market. After in-depth business research, the *Clarity Score* is calculated from online content based on three components: the sentiments of the chosen drivers, the overall rating, and the volume of online content.

In *Clarity 1.0*, these three components are assigned percentile scores, as discussed in the System Overview section. Because the percentile values are between 0 and 100, the final *Clarity score* calculated on the weighted sum of these components are also bounded between 0 and 100, which can be scaled to a 10-point rating system as shown in Figure 1. However, because only a small number of products are compared within a competing market for refined insights, the percentile score computed within this set is not meaningful due to insufficient data in the distribution for a single market. To solve this problem, in *Clarity 1.0*, all the products are placed in the same pool to calculate the percentile scores. Moreover, an exponential decay function over past time frames is introduced to account for data recencies.

Because of the application of percentiles and the time decay function, the *Clarity Score* calculated was often difficult for end users to understand. User feedback was highly positive in the competitive ranking; however, in order to ensure a broad understanding of the core drivers and how to influence and improve the *Clarity score*, the approach required adjustments to match user-identified needs to fit seamlessly into workflows. In order to improve explainablilty and thus ultimate impact of *Clarity*, a new *Clarity*

Much easier to use than the competition and you will see nearly immediate results . (Ease of use - 0.98308)
Much easier to use than the competition and you will see nearly immediate results . (Performance and efficiency - 0.53858)
Much easier to use than the competition and you will see nearly immediate results . (Product competitiveness - 0.50202)

FIGURE 6 Example of topic classification using the attention mechanism

Score methodology is developed which we will also refer to as *Clarity Score 2.0*. In the new methodology, the three components, the sentiments on different topics, the overall rating, and the content volume, are still considered. However, instead of using percentiles and decay functions, a more rigorous approach based on Bayesian statistics is used.

Bayesian sentiment

Same as the previous approach, the IBM Watson NLU service is used for sentiment analysis. The output of the Watson NLU sentiment model is between -1 and 1, representing negative and positive sentiments. A simple approach to obtain a monthly sentiment value is to average the sentiments of all online content for the specific product within that month.

However, due to business reasons, many of the products considered do not have a large volume of content, and in some cases, a few products might not have online content at all every month. Taking average on a small volume of online content or assigning 0 to products with no content does not reflect the true sentiment on these products. Instead of using the average sentiment, we can consider the online sentiment as a probabilistic process. Let $P(s|D)$ be the probability distribution of the sentiment value s conditioned on the data D, then the sentiment value for that month is the argmax of $P(s|D)$ based on maximum likelihood. The distribution $P(s|D)$ can be updated every month by incorporating new data using the Bayesian method, such that at the beginning of each month, there is a prior assumption about the overall sentiment value. Based on the actual sentiments of public commentary in the month, the prior assumption is updated to obtain a posterior sentiment.

To apply Bayesian update to the sentiment values, sentences with a neutral sentiment are removed because they don't contribute to the overall sentiments. When only considering the numbers of positive and negative sentences, the distribution conditioned on the normalized average sentiment value s is the Bernoulli distribution, the corresponding conjugate distribution is the Beta distribution with parameters a and b. The values of a and b are updated based on the counts of positive and negative posts. Mathematically, $a_{\text{posterior}} = a_{\text{prior}} + n_{\text{positive}}$, and $b_{\text{posterior}} = b_{\text{prior}} + n_{\text{negative}}$, where n_{positive} is the number of positive sentiments and n_{negative} is the number of negative sentiments.

The prior values of a and b are assumptions on the numbers of positive and negative commentary. For each month, these values can be taken from the posterior values in the previous month. However, a Bayesian update process implemented this way aggregates all the content before the current month, such that an online post from 2 years ago would be considered in the same way as a piece of content posted this month. In order to avoid the propagation of online content indifferently over time, the values of a and b are re-scaled at the beginning of each month. Because the typical values of a and b are 1 for a general Beta prior, the values of a and b are re-scaled such that the sum of a and b is 2. Thus, $a_{\text{rescaled}} = 2a/(a+b)$, and $b_{\text{rescaled}} = 2b/(a+b)$.

Using this approach, we are able to obtain a posterior average sentiment value for each month. When the content volume is large, the posterior sentiment value is the same as the average sentiment value. When the content volume is small, the posterior sentiment is shifted based on the value from the previous month. Because the values of a and b are known, the uncertainties in the overall sentiment value can be calculated by using the 95% confidence interval.

Bayesian rating

Similarly, the rating can be calculated using a similar Bayesian method. However, while the sentiment values can be simplified to be positive or negative based on the Bernoulli distribution, the typical 5 star rating is not trivial. In order to apply a Bayesian update on ratings, we assume that the distribution of ratings follows the Binomial distribution, $B(4, p)$, where p is the average rating, and the 5 possible outcomes, [0, 1, 2, 3, 4], of $B(4, p)$ correspond to the 1 to 5 stars.

By assuming the data distribution to be Binomial distribution, the corresponding conjugate prior is also the Beta distribution; same as the case for sentiment values. Thus, rather than representing the number of positive and negative posts, the value of a represent the number of stars in the rating. The set of ratings, [1, 2, 3, 4, 5], is mapped to the outcome of the Binomial distribution, which is [0, 1, 2, 3, 4]. Thus, a rating of 2 corresponds to a 1 in the Binomial distribution, thus contributing 1 to the value of a and 3 to the value of b in the Bayesian update.

Since the rating is treated in an analogous way as the sentiment value, the prior parameters a and b are also rescaled from the previous month. Rather than rescaling a and b to sum to 2, they are summed to 8 instead, due to the fact that the maximum values of a and b are 4 from the Binomial distribution.

Scaled post counts

Lastly, the content counts are rescaled using a nonlinear function, rather than using the percentile from the pool

of posts from all products. It should be noted that rather than comparing the absolute difference between content counts, it is more important to compare the relative ratios. For example, even though the post counts are differed by 100, products with post counts of 100 and 200, are compared differently for products with post counts of 800 and 900. For the former case, 200 posts is twice the amount of 100 posts, while the relative ratio between 800 and 900 is smaller.

Since the ratios between post counts are compared, it is thus appropriate to use the logarithmic function. Thus, the difference between 200 and 100 posts are then log(200)-log(100) = log(2), while the difference between 900 and 800 are log(900)-log(800) = log(9)-log(8), which is much smaller for the latter case. However, as mentioned previously, many products do not have posts every month. Applying a logarithm on 0 leads to an undefined value. Thus, rather than using logarithm, a log1p function is used. The log1p function is defined as $\log1p(x) = \log(1 + x)$.

Lastly, the range of log1p function is $[0, \infty)$ for a non-negative number of posts. However, in order to obtain a bounded *Clarity Score*, the scaling function for post counts must be bounded above as well. One way to do this is to divide the log1p function by itself when the number of posts is large. Thus, the final scaling function for the post counts is: $f(n) = \log1p(n\text{-}n_{\min})/(\log1p(n_{\max}\text{-}n_{\min})+c)$, where n is the number of posts for the product, $n_{\min}$ and $n_{\max}$ are the minimum and maximum number of posts for all the products in the competing market, and c is a small constant. Based on our experimentation, c is chosen to be $1e^{-5}$.

Overall Clarity Score 2.0

Similar to the previous approach, the *Clarity Score* is the weighted sum of the three components: Bayesian sentiment, Bayesian rating, and scaled post count. Since the dependence in the past has been considered in the Bayesian updates of sentiment values and ratings, a decay function is no longer necessary. By removing the complexity of the percentile method and the decay function, the contributions of the three components of the *Clarity Score* is easier to identify.

As discussed in the *Improved NLP Model* section, the output of the NLP model is expanded to cover 14 topics to provide more detailed information about the online content. However, because products are different across different markets, the weights on these topics are considered differently in different market. To account for the importance of each topic, the weights on the topics are calculated based on the frequencies of these topics within the same market. In *Clarity* 1.0, 5 drivers are considered for the calculation of the *Clarity Score*. Similarly in *Clarity* 2.0, only the top 5 topics based on the weights are selected as drivers. Thus, the sentiment values from Watson NLU for each product is first aggregated on each of the 5 drivers, and then summed to the overall sentiment value based on the weights.

TABLE 1 Model performance on one of the topics, product features and functionalities

Model	Precision	Recall	F1 Score
Neural network on sentence	0.40	0.32	0.36
Neural network on words	0.56	0.40	0.47
Neural network based on attention	0.56	0.56	0.56
Human performance	0.75	0.56	0.64

Discussion – performance and beyond

As *Clarity* becomes more essential in product analysis, we have made significant improvements to the user experience. Particularly, the ranking score, or *Clarity Score*, is revised to provide a clearer comparison between products in the same space. At the same time, the NLP models are improved to extract more accurate insights from the user content.

NLP performance

As discussed in the *Improved NLP Model* section, the newly developed attention-based topic classification model associates keywords in the online content and the corresponding topic. This is a crucial feature to enable interpretability that is typically absent in conventional black box machine learning algorithms.

Another significance of the improved NLP algorithm is the overall performance in topic classification. In order to benchmark the performance, a ground truth set with 200 sentences was established. The performances of the new algorithms in *Clarity* 2.0 and human labelers are compared on this ground truth set.

As reflected in the F1 scores in Table 1, the overall performance is progressively better from the simple sentence based neural network to the more complex word based neural network, and finally to the more sophisticated attention based neural network. The main improvement of the attention based neural network is to consider the importance of keywords as attention weights explicitly when assigning topics to input text. Because the weights are dependent on the presence of other words, contextual information is partially considered.

Precision, recall, and F1 score are compared for the models described in the Improved NLP Model in Clarity 2.0 section. It should be noted that the human

performance is relative low compared to typical academic datasets. The reason is that practical NLP applications are typically ambiguous, which results in more noise in the results and leads to the overall lower score compared to well-refined academic datasets.

Clarity Score performance

Due to the explainability issues with the former *Clarity Score* methodology, the *Clarity Score* algorithm was improved as described in the previous section. With the new implementation, the ranking is now more transparent as it is related directly to the three components, post counts, ratings, and over sentiment scores, without the unnecessary percentile scaling and the correlation with past data due to an exponential time decay function. Although the new *Clarity Score* methodology is easier to explain, it must be validated against the same metrics, including the Net Promoter Score (NPS) and the Gartner Magic Quadrant, as described in the previous *Clarity* 1.0 paper.

Top: comparison between the old and new Clarity Scores. Bottom: comparison between the Clarity Scores and NPS.

Comparison between NPS, *Clarity Score* 1.0, and *Clarity Score* 2.0

First, the *Clarity Score* 2.0 is compared to the *Clarity Score* 1.0, as shown in Figure 7. In the figure, the Clarity Score 2.0 is almost linearly correlated with the *Clarity Score* 1.0, implying that the rankings of the offerings are preserved. As explained in (Usmani et al. 2020), the *Clarity Score* 1.0 was a good metric based on in-lab and expert evaluations. The strong correlation with the *Clarity Score* 1.0 allows the new *Clarity Score* 2.0 to provide insightful information for the users.

Next, both the old and new *Clarity Scores* are compared to the NPS. The *Net Promoter Score* is a metric used in customer satisfaction research. The NPS is calculated based on responses to the question: How likely is it that you would recommend this product? and the answer is based on a 0 to 10 scale. People who have responded with a score of 9 or 10 are called promoters, those who have responded with scores between 7 and 8 are called passives, and the ones that have responded with scores between 0 and 6 are called detractors. The NPS is the difference between the percentage of promoters and detractors. For comparison with the *Clarity Score*, the NPS is rescaled to the same range from 0 to 10.

As shown in the bottom plot of Figure 7, both the old and new *Clarity Scores* are linearly correlated to the NPS. However, the *Clarity Score* 2.0 shows a stronger correlation as the points are closer to the diagonal line, and the spread of the points is much smaller compared to *Clarity Score* 1.0, indicating that the new score is more reflective of the customer's satisfaction.

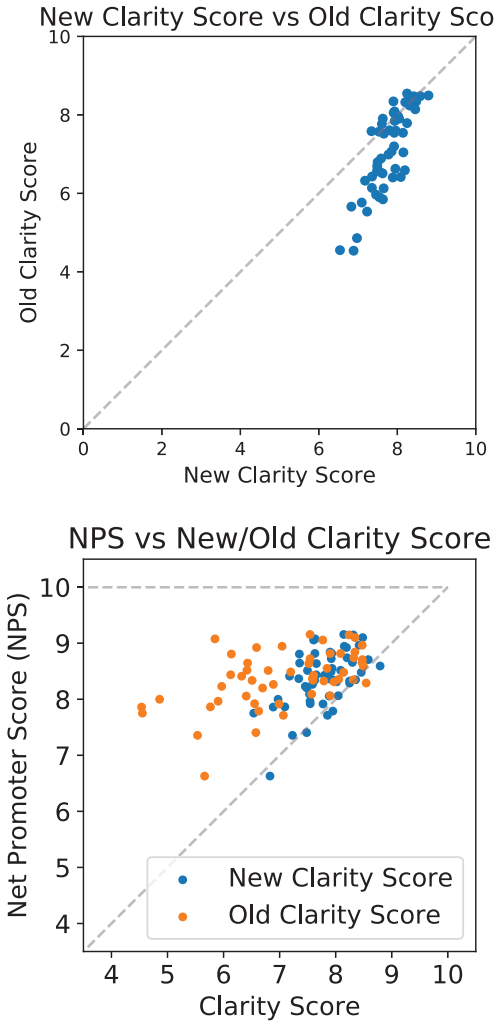

FIGURE 7 Comparison between NPS, Clarity Score 1.0, and Clarity Score 2.0

Comparison between the Clarity Score 2.0 and Gartner Magic Quadrants (MQ)

The IT consulting firm *Gartner* periodically produces a series of market research reports where they rate vendors according upon two criteria: completeness of vision and ability to execute (Gartner, Inc. n.d.). Each of these reports include a 2×2 matrix chart similar to the one depicted in Figure 8.

Vendors with both a high completeness of vision and a high ability to execute are called *leaders* (vendors A and E in Figure 8) whereas vendors with low scores in both

FIGURE 8 Gartner Magic Quadrant

dimensions are called *niche players* (vendor B). Vendors with a high completeness of vision but with a low score in the ability to execute are called *visionaries* (vendors C and G) and the vendors with a poor completeness of vision but good ability to execute are called *challengers* (vendors D and F).

Similar to the analysis in the previous paper, we analyzed 5 randomly selected markets provided by Gartner and noted the similarities and differences with the *Clarity Scores*. Given a Gartner Magic Quadrant report for market M_i, we identified the vendors $V^{Mi} = \{V_1^{Mi}, V_2^{Mi}, \ldots, V_n^{Mi}\}$ in that report and we then identified the products $P^{Vj,Mi} = \{P_1^{Vj,Mi}, P_2^{Vj,Mi}, \ldots, P_m^{Vj,Mi}\}$ such that $P_K^{Vj,Mi}$ is a product in the market M_i provided by the vendor V_j^{Mi}. We then compared the Clarity scores for the products in P_{Vj}^{Mi} for which we had data.

As shown in Table 2, the magic quadrant market labels align with the Clarity scores for most cases. A high *Clarity Score* indicates that the vendor is highly competitive, which is labeled as leader in the Gartner Magic Quadrant. A relatively lower *Clarity Score* implies that the vendor is not competitive in the market, which is labeled as niche player by the Gartner Magic Quadrant. The immediately competitive vendors are labeled as visionaries or challengers, which have values between niche players and leaders. However, there is an outlier for M_1 = Data Science Machine Learning Platform, where V_5^{MI} is labeled as leader, but as the smallest *Clarity Score* compared to the other vendors.

Broader use cases

In addition to improving explainability and taking steps to ensure insights resonated with and prompted action for end users, the team also explored new use cases that surfaced as a result of the success of the *Clarity* insight engine.

Some use cases were prompted by a desire to make similar analyses available from net new data sources, while others were inspired by *Clarity*'s utility for product owners and prompted work flow re-imagination for other roles across the enterprise leveraging similar analytical approaches. New use cases were sourced from user suggestions through automated feedback mechanisms, co-creation oriented design thinking sessions with existing users, as well as ideas developed within the product team. New use cases under development are at varying degrees of maturity – the most advanced is a version of *Clarity* that focuses on industry analyst content.

The Clarity for Industry Analysts use case, currently at Minimum Viable Product (MVP) stage, ingests and analyzes analyst report data to support both product managers and analyst relations professionals. The Clarity system produces insights from analysis of thousands of reports, blogs, presentations, and articles from top industry analyst firms, similarly leveraging an algorithm built around Watson NLU to calculate sentiment and share of voice analyses for IBM and our competitors. Reports are grouped based on IBM alignment of analysts to market areas, which then align to IBM's internal product taxonomy. By understanding sentiment and share of voice at a firm, analyst, and market level, analyst relations professionals can optimize their time, and product managers can better understand the assessment of their products from third party organizations.

Beyond industry analyst reports, the Clarity team continues to explore a variety of different data sources and approaches to provide competitive insights to existing and adjacent user bases. These include combining financial performance data and unstructured text from discussion forums in online support communities to inform product and portfolio performance, and even goes beyond product performance in experimenting in assessment of IBM's competitive standing in sustainability and responsibility initiatives.

Conclusion

In this paper, we considered the problem of comparing products in a marketplace automatically from online content. This is an important business activity that marketers, sellers and product managers conduct regularly. Unfortunately, it is also very time consuming and costly which can

TABLE 2 Comparison of ratings provided by Gartner vs. Clarity Score

Market	Vendor	Product	Gartner Q	*Clarity* score
M_1 = Data Science Machine Learning Platform	V_1^{M1}	$P_1^{V1, M1}$	Visionary	8.79
	V_2^{M1}	$P_1^{V2, M1}$	Visionary	8.48
	V_3^{M1}	$P_1^{V3, M1}$	Challenger	7.93
	V_4^{M1}	$P_1^{V4, M1}$	Challenger	7.91
	V_5^{M1}	$P_1^{V5, M1}$	Leader	7.34
M_2 = Data Management Solutions for Analytics	V_1^{M2}	$P_1^{V1, M2}$	Leader	8.58
	V_2^{M2}	$P_1^{V2, M2}$	Leader	8.46
	V_3^{M2}	$P_1^{V3, M2}$	Leader	8.19
	V_4^{M2}	$P_1^{V4, M2}$	Leader	7.96
	V_5^{M2}	$P_1^{V5, M2}$	Leader	7.88
M_3 = Management Solutions for Analytics	V_1^{M3}	$P_1^{V1, M3}$	Leader	7.62
	V_2^{M3}	$P_1^{V2, M3}$	Leader	7.53
	V_3^{M3}	$P_1^{V3, M3}$	Leader	7.36
M_4 = Operational Database Management Systems	V_1^{M4}	$P_1^{V1, M4}$	Leader	8.58
	V_2^{M4}	$P_2^{V2, M4}$	Leader	8.46
	V_3^{M4}	$P_3^{V3, M4}$	Leader	8.19
M_5 = Analytics and Business Intelligence Platform	V_1^{M5}	$P_1^{V1, M5}$	Leader	8.79
	V_1^{M5}	$P_2^{V1, M5}$	Leader	8.25
	V_2^{M5}	$P_1^{V2, M5}$	Niche Player	7.92
	V_3^{M5}	$P_1^{V3, M5}$	Niche Player	7.85

be particularly challenging for businesses with large product portfolios and fast-changing customer environment.

To address this business need, in an earlier work which appeared at IAAI-20, the 2020 Innovative Applications of Artificial Intelligence (Usmani et al. 2020), we introduced a deployed system, called *Clarity*, which analyzes the competitive landscape of products in a marketplace continuously as data gets updated over time. This paper improves on the system with an enhanced NLP model to detect product reviews and a new scoring method to rank them which is closer to user's expectation. We ran experiments to validate *Clarity*'s usefulness and scalability. *Clarity* has thus proven to be an excellent example of an AI-based system that has been integrated and reused in various applications such as product pricing recommendation, and talent management and has performed well in critical business activities. Further expansion of *Clarity* within IBM will explore new use cases for different business roles or data source types.

ACKNOWLEDGMENTS

This research was supported by IBM's Chief Analytics Office, with support and partnership from the Analyst Relations team, Market Development and Insights, and Global Offering Management organization. Individuals who contributed to the development and scale of *Clarity* 1.0 include Amir Sabet Sarvestani, Casey Klippel, Jeanine Chong, Prajjalita Dey, Timothy Lee, Sunny Sharma, and Alex Vassilenko. Individuals that contributed to the development and scale of *Clarity* 2.0 include Marisela Cadenas, Jon Smith, Nisarg Patel, Alexandra Jones, Louis Monteagudo, Daniel Mandel, Eldhose Shadi, Alejandro Corbellini, and Gregorio Gerardi.

REFERENCES

Afful-Dadzie, E., S. Nabareseh, Z. K. Oplatkova, and P. Klimek. 2014. "Enterprise Competitive Analysis and Consumer Sentiments on Social Media - Insights From Telecommunication Companies." In Proceedings of 3rd International Conference on Data Management Technologies and Applications - Volume 1: DATA, 22–32.

Amarouche, K., H. Benbrahim, and I. Kassou. 2015. "Product Opinion Mining for Competitive Intelligence. Procedia Computer Science 73:358–65." International Conference on Advanced Wireless Information and Communication Technologies (AWICT 2015).

Bhatt, D. A., K. E. Mcneil, and N. A. Patel. 2014. "Time- Based Sentiment Analysis for Product And Service Features." *In Journal Software* 9(2), https://patents.google.com/patent/US9177554B2/en.

Bostock, M., V. Ogievetsky, and J. Heer. 2011. "D3: Data-driven Documents." IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis).

Gartner, Inc n.d. Gartner Magic Quadrant Research Methodology. https://www.gartner.com/en/research/methodologies/magic-quadrants-research. Accessed: 2019-09-05.

Joung, J., K. Jung, S. Ko, and K. Kim. 2018. "Customer Complaints Analysis Using Text Mining and Outcome - Driven Innovation Method for Market-Oriented Product Development." *Sustainability* 11(1): 1–14.

Srivastava, B., M. Chetlur, S. Gupta, M. Vasa, and K. Visweswariah. 2018. "Effective Business Development for In-Market it Innovations with Industry-Driven Api Composition." In *Data Science Landscape. Studies in Big Data*, Vol 38. Springer.

Usmani, S., M. Bernagozzi, Y. Huang, M. Morales, A. S. Sarvestani, and B. Srivastava. 2020. "Clarity: Data-driven Automatic Assessment of Product Competitiveness." In *AAAI*.

AUTHOR BIOGRAPHIES

Yufeng Huang is an adjunct professor at Sacred Heart University in Connecticut, USA. He was the technical lead of the *Clarity* project and a lead data scientist at the Chief Analytics Office at IBM. He received his BS degree and MS degree from Rensselaer Polytechnic Institute, and his PhD from California Institute of Technology.

Mariana Bernagozzi is a lead data scientist at IBM Global Business Services. Prior to joining IBM, she worked as a software engineer both in industry and academia, primarily with embedded systems, gaming, and multimedia. She holds a master's degree in Computer Science from Universidad Nacional de La Plata.

Michelle Morales is a lead data scientist with IBM Global Business Services, where she builds AI solutions for IBM clients. Before joining IBM, Michelle received her PhD in Computational Linguistics and conducted research at the intersection of linguistics, computer science, and psychology.

Sheema Usmani is a data scientist at IBM Global Business Services. She earned her Master's in Computational and Mathematical Engineering from Stanford University. Her undergraduate degree is in Applied Mathematics from IIT BHU, India.

Biplav Srivastava is a professor of computer science at the AI Institute at the University of South Carolina. Previously, he was at IBM for nearly two decades in the roles of a Research Scientist, Distinguished Data Scientist and Master Inventor. Biplav is an ACM Distinguished Scientist, AAAI Senior Member, IEEE Senior Member and AAAS Leshner Fellow for Public Engagement on AI (2020-2021).

Michelle Mullins is a Senior Managing Consultant in IBM's Chief Analytics Office, leading a portfolio of Offering and Marketing Analytics projects as well as Social Impact work within the organization. She has a BS in Government and Global Studies from the College of William and Mary, and an MBA from NYU Stern School of Business in Strategy, Social Impact and Innovation.

How to cite this article: Huang, Y., M. Bernagozzi, M. Morales, S. Usmani, B. Srivastava, and M. Mullins. 2021. "Clarity 2.0: Improved Assessment of Product Competitiveness from Online Content." *AI Magazine* 42: 59–70. https://doi.org/10.1609/aaai.12006.

It is the generosity and loyalty of our members that enable us to continue to provide the best possible service to the AI community and promote and further the science of artificial intelligence by sustaining the many and varied programs that AAAI provides. AAAI invites all members and other interested parties to consider a gift to help support the dozens of programs that AAAI currently sponsors.
For more information about the Gift Program, please use the donate option when renewing your membership, or write to us at donate21@ aaai.org.

AAAI is a 501c3 charitable organization. Your contribution may be tax deductible.